MIDDLE-END SOLUTION TO ROBUST SPEECH RECOGNITION

FIELD OF INVENTION:

[0001] This invention relates to speech recognition and more particularly to Signal-to-Noise Ratio (SNR) dependent decoding and weighted Viterbi recognition.

BACKGROUND OF INVENTION:

[0002] A technique of time-varying SNR dependent coding for increased communication channel robustness is described by A. Bernard ,one of the inventors herein, and A. Alwan in "Joint channel decoding –Viterbi Recognition for Wireless Applications ", in Proceedings of Eurospeech, Sebt. 2001, vol. 4, pp. 2703-6; A. Bernard, X. Liu, R. Wesel and A. Alwan in "Speech Transmission Using Rate-Compatable Trellis codes and Embedded Source Coding," IEEE Transactions on Communications, vol. 50, no. 2, pp 309-320, Feb. 2002.; and A. Bernand in "Source and Channel Coding for Speech and Remote Speech Recognition," Ph.D. thesis, University of California, Los Angeles, 2001.

[0003] For channel and acoustic robustness is described by X. Cui, A. Bernard, and A. Alwan in "A Noise-robust ASR back-end technique based on Weighted Viterbi Recognition, " in Proceedings of Eurospeech, September 2003, pp. 2169-72.

[0004] Speech recognizers compare the incoming speech to speech models such as Hidden Markov Models HMMs to identify or recognize speech. Typical speech recognizers combine the likelihoods of the recognition features of each speech frame with equal importance to provide the overall likelihood of observing the sequence of feature vectors. Typically robustness in speech recognition is dealt with either at the front end (by cleaning up the features) or at the back end (by adapting the acoustic model to the particular acoustic noise and channel environment).

[0005] Such classic recognizers fail to differentiate between the particular importance of each individual frame, which can significantly reduce recognition performance when the importance of each frame can be quantitatively estimated into a weighted recognition mechanism.

SUMMARY OF INVENTION:

[0006] In accordance with one embodiment of the present invention a procedure for performing speech recognition which can integrate, besides the usual speech recognition feature vector, information regarding the importance of each feature vector (or even frequency band within the feature vector). Applicant's solution leaves both the acoustic features and models intact and only modifies the weighting formula in the combination of the individual frame likelihoods.

[0007] In accordance with an embodiment of the present invention a method for performing time and frequency SNR dependent weighting in speech recognition includes for each period t estimating the SNR to get time and frequency SNR information $\eta_{t,f;}$; calculating the time and frequency weighting to get $\gamma_{t,f;}$ performing the back and forth weighted time varying DCT transformation matrix computation MG_tM^{-1} to get T_t; providing the transformation matrix computation T_t and the original MFCC feature o_t that contains the information about the SNR to a recognizer including the Viterbi decoding; and performing weighted Viterbi recognition $b_j(o_t)$.

DESCRIPTION OF DRAWING:

[0008] Figure 1 is an illustration of the Viterbi algorithm for HMM speech where the vertical dimension represents the state and the horizontal dimension represents the frames of speech (i.e. time).

[0009] Figure 2 is a block diagram of time and frequency SNR dependent weighted Viterbi recognition.

[0010] Figure 3 illustrates the performance of t-WVR back-end on the Aurora-2 database for different SNRs.

DESCRIPTION OF PREFERRED EMBODIMENT:

Review of Time Weighted Viterbi Recognition

[0011] In general, there are two related approaches to solve the temporal alignment problem with HMM speech recognition. The first is the application of dynamic programming or Viterbi decoding, and the second id the more general forward/backward algorithm. The Viterbi algorithm (essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation) is typically used for segmentation and recognition and the forward/backward for training. See for the Viterbi algorithm G.D. Fornay, " The Viterbi algorithm, " IEEE Transactions on Communications, vol. 61, no. 3, pp. 268-278, April 1973.

[0012] The Viterbi algorithm finds the state sequence Q that maximizes the probability $P*$ observing the features sequence (O=o_1,...o_t) given the acoustic model λ

$$P* = \max_{\text{All } Q} P\,(\mathrm{Q}, O \,|\, \lambda). \qquad (1)$$

[0013] In order to calculate $P*$ for a given model λ , we define the metric $\varphi_j(\mathrm{t})$, which represents the maximum likelihood of observing the features sequence (O=o_1,...o_t)

given that we are in state j at time t. Based on dynamic programming, this partial likelihood can be computed efficiently using the following recursion

$$\varphi_j(t) = \max_i \{ \varphi_j(t-1)a_{ij} \} b_j(o_t). \qquad (2)$$

[0014] The maximum likelihood $P^*(O \mid \lambda)$ s then given by $P^* (O \mid \lambda) = \max_j \{ \varphi_j(T) \}$.

[0015] The recursion (2) forms the basis of the Viterbi Algorithm (VA) whose idea is that there is only one "best" path to state *j* at time *t*.

[0016] As shown in Figure 1, this algorithm can be visualized as finding the best path through a trellis where the vertical dimension represents the states of the HMM and the horizontal dimension represents the frames of speech (i.e. time).

Time Weighted Viterbi Recognition (WVR)

[0017] In speech recognition, the quality of speech features can depend on many factors: acoustic noise, microphone quality, quality of communication, etc. The weighted Viterbi recognizer (WVR), presented in the "Joint Channel Decoding-Viterbi Recognion for Wireless Applications,"cited above, modifies the Viterbi algorithm (VA) to take into account the quality of the feature.

[0018] The time-varying quality γ_t of the feature vector at time *t* is inserted in the VA by raising the probability $b_j(o_t)$to the power γ_t to obtain the following state metrics update equation:

$$\varphi_{j,t} = \max_i [\varphi_{i,t-1}\, a_{ij}] [b_j (o_t)]^{\gamma_t}. \qquad (3)$$

where $\varphi_{j,t}$ is the state metric for state *j* at time *t* and a_{ij} is the state transition metric. Such weighting has the advantage of becoming a simple multiplication of $\log (b_j(o_t))$ by γ_t in the logarithmic domain often used for scaling purposes. Furthermore, note that if one is certain about the received feature, $\gamma_t = 1$ and equation 3 is equivalent to equation. 2. On the other hand, if the decoded feature is unreliable, $\gamma_t = 0$ and the probability of observing the feature given the HMM state model $b_j(o_t)$ is discarded in the VA recursive step.

[0019] Under the hypothesis of a diagonal covariance matrix Σ, the overall probability $b_j(o_t)$ can be computed as the product of the probabilities of observing each individual feature. The weighted recursive formula (equation 3) can include individual weighting factors $\gamma_{t,t}$ for each of the N_F front-end features.

$$\varphi_{j,t} = \max\,[\varphi_{i,t-1}\, a_{ij}] \prod_{k=1}^{N_F} [b_j\,(o_t)]^{\gamma}{}_{k,.t..} \qquad (4)$$

where k indicates the dimension of the feature observed.

Time *and frequency* WVR

[0020] In accordance with the present invention we provide an extension to the time-only weighted recognition presented in equation3. First, we present how we can use both time and frequency weighting. Second, we present how the weighting coefficients can be obtained.

Time and frequency weighting

[0021] With time weighting only, the insertion of the weighting coefficient in the overall likelihood computation could be performed *after* the probability $b_j\,(o_t)$ had been computed by raising it to the power γ_t, using $\tilde{b}_j(o_t) = [b_j\,(o_t)]^{\gamma}{}_{t.}$

[0022] In order to perform time and frequency SNR dependent weighting, we need to change the way the probability $b_j(o_t)$ is computed. Normally, the probability of observing the N_F -dimensional feature vector o_t in the j^{th} state is computed as follows,

$$b_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}[\Sigma]}} e^{-\frac{1}{2}(o_t-\mu)'\Sigma^{-1}(o_t-\mu)}, \qquad (5)$$

where N_M is the number of mixture components, w_m is the mixture weight, and the parameters of the multivariate Gaussian mixture are its mean vector μ and covariance matrix Σ.

[0023] In order to simplify notation, we should only note that $\log(b_j(o_t))$ is proportional to a weighted sum of the cepstral distance between the observed feature and the cepstral mean (o_t-μ), where the weighting coefficients are based on the inverse covariance matrix (Σ^{-1}),

$$\log(b_j(o_t)) \quad \infty\, (ot-\mu)' \ \Sigma^{-1} (o_t-\mu\ ). \qquad (6)$$

[0024] Remember that the N_F -dimensional cepstral feature o_t is obtained by performing the Discrete Cosine Transform (DCT) on the N_S- dimensional log Mel spectrum (*S*). Mathematically, if the N_S x N_F dimensional matrix *M* represents the DCT transformation matrix, then we have o_t = MS. Reciprocally, we have S= M^{-1} o_t where M^{-1} (N_S x N_F) represent the inverse DCT matrix.

[0025] Since usually the frequency weighting coefficients we have at hand will be in the log spectral domain (whether linear or Mel spectrum scale is not important) and not in the cepstral domain, we use the inverse DCT matrix S= M^{-1} to transform the cepstral distance (o_t-μ) into a spectral distance. Once in the spectral domain, time and frequency weighting can be applied by means of a time-varying diagonal matrix G_t which represents the weighting coefficients $\gamma_{t,f}$,

$$G_t = \mathrm{diag}(\gamma_{t,f}) \quad . \qquad (7)$$

[0026] Finally, once the weighting has been performed, we can go back to the spectral domain by performing the forward DCT operation. All together, the time and spectral frequency weighting operation on the cepstral distance d= (o_t -μ) becomes

$$\tilde{d} = MG_tM^{-1}\,(o_t - \mu) \qquad (8)$$

[0027] With this notation, the weighted probability of observing the feature becomes

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}[\Sigma]}} e^{-\frac{1}{2}(o_t-\mu)'(MG_tM^{-1})'\Sigma^{-1}(MG_tM^{-1})(o_t-\mu)} \qquad (9)$$

which can be rewritten using a back-and-forth weighted time-varying transformation matrix $T_t = MG_tM^{-1}$ as

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}[\Sigma]}} \ \mu) \quad e^{-\frac{1}{2}(o_t-\mu)'(T_t')\Sigma^{-1}(T_t)(o_t-\mu)} \ , \qquad (10)$$

which can also resemble the unweighted equation 5 with a new inverse covariance matrix

$\tilde{\Sigma}^{-1} = T_t' \, \Sigma^{-1} \, T_{t,}$

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}[\Sigma]}} e^{-\frac{1}{2}(o_t-\mu)'\tilde{\Sigma}^{-1}{}_t(o_t-\mu)} \qquad (11)$$

[0028] To conclude this part on time and frequency weighting, note that time weighting only is a special case of time and frequency weighting where $G_{t=}\gamma_t.I$ where I is the identity matrix, which also means that the weighting is the same for all the frequencies.

Determining the Weighting Coefficients

[0029] In order to have the system performing SNR dependent decoding, we first need a time and frequency SNR evaluation. In the special case presented above, the time frequency scale is the frame based (every 10 ms) and the frequency scale is the Mel frequency scale, which divides the narrowband speech spectrum (0-4kHz) in 25 non-uniform bandwidth frequency bins.

[0030] In that specific case, the time and frequency SNR evaluation we are using for the purpose of evaluating the presented technique is that of the ETSI Distributed Speech Recognition standard [6] which evaluates the SNR in the time and frequency domain for spectral subtraction purposes. See ETSI STQ-Aurora DSR Working Group, "Extended Advanced Front-End (xafe) Algorithm Description," Tech. Rep., ETSI, March 2003.

[0031] Regardless of the technique used to obtain such time and frequency dependent SNR estimate, we decide to refer to such value as $\eta_{t,f}$. $\eta_{t,f}$ is the SNR at frequency f at t time. The weighting coefficient $\gamma_{t,f}$ can be obtained by performing any function which will monotonically map the values taken by the SNR evaluation (logarithmic or linear) to the interval [0,1] of the values that can be taken by the weighting coefficients $\gamma_{t,f}$. In other words, we have

$$\gamma_{t,f}=f(\eta_{t,f}). \qquad (12)$$

[0032] One particular instantiation of equation 12 is using a Wiener filter type equation applied on the linear SNR estimate to obtain,

$$\gamma_{t,f} = \frac{\sqrt{\eta_{t,f}}}{1+\sqrt{\eta_{t,f}}} \quad ,$$

which guarantees that $\gamma_{t,f}$ is equal to 0 when $\eta_{t,f}$=0 and $\gamma_{t,f}$ approaches 1 when $\eta_{t,f}$ is large.

[0033] Figure 2 illustrates the block diagram for the time and frequency weighted Viterbi recognition algorithm. When you have speech (speech frame t) the first step 21 is to estimate the SNR to get $\eta_{t,f}$. Then the weighting is calculated to get $\gamma_{t,f}$ at step 23. Then the transform matrix computation at step 25 is performed. This is the MG_tM^{-1} to get Tt . The next step is Viterbi decoding at step 27 to get $b_j(o_t)$. Here the original MFCC feature o_t is sent to the recognizer. The original feature contains the information about the SNR.

Performance evaluation

Experimental conditions

[0034] We used the standard Aurora-2 testing procedure, which averages recognition performance over 10 different noise conditions (two with channel mismatch in Test C) at 5 different SNR levels (20dB, 15dB, 10dB, 5dB and 0dB).

[0035] As a reminder, performance is established using the following configuration: a 21-dimensional feature vector (16 Mel frequency cepstral coefficients (MFCC) features with 1st order derivative) extracted every 10 ms and 16 states word HMM models with 20 Gaussian mixtures per state.

Performance of time-WVR algorithm

[0036] Figure 3 summarizes the performance of time-WVR algorithm on the Aurora-2 database. As expected, the t-WVR algorithm improves recognition accuracies mainly in the medium SNR range. Indeed, it is in the medium SNR range that the frames distinction that can be obtained by performing SNR dependent weighting is the most useful. At low (resp. high) SNR range most features are already usually bad (good).

[0037] In accordance with the present invention the weighting function can be applied in the logarithmic domain using a simple multiplicative operation. The weighting coefficient can be the output of many different important estimation mechanisms, including a frame SNR estimation, a pronunciation probability estimation, a transmission over a noisy communication channel reliability estimation, etc.

[0038] Although preferred embodiments have been described, it will be apparent to those skilled in the art that various modifications, additions, substitutions and the like can be made without departing from the spirit of the invention and these are therefore considered to be within the scope of the invention as defined in the following claims.